

What is claimed is:

1. A method comprising
getting an object header from an object, and
5 checking from the object header a result of a first time type checking
between a class of the object and a target class specified by a hotspot in the first
time type checking.
2. The method of claim 1 further comprising
10 determining whether the object header comprises an indicator that is
asserted to indicate a first time type checking success between the object class and
the target class associated with the indicator.
3. The method of claim 1 further comprising
15 determining whether the object header comprises an indicator that is
deasserted to indicate a first time type checking failure between the object class
and the target class associated with the indicator.
4. The method of claim 1 further comprising
20 skipping a second time type checking between the object class and the
target class, in response to determining that the object header comprises an
Indicator that is asserted to indicate a first time type checking success.

5. The method of claim 1 further comprising
performing a second time type checking between the object class and the
target class, in response to determining that the object header comprises an
indicator that is deasserted to indicate a first time type checking failure.

5

6. The method of claim 1 further comprising
detecting the hotspot in the first time type checking by dynamic profiling.

7. A system, comprising
10 a processor to get an object header from an object, and obtain from the
object header a result of a first time type checking at a hotspot between a class of
the object and a target class specified by the hotspot; and
a memory to save the target class.

15 8. The system of claim 7, wherein the processor further to
determine that the first time type checking at the hotspot is successful, in
response to detecting that the object header comprises an indicator associated with
the target class that has a first logic value.

20 9. The system of claim 7, wherein the processor further to
perform a second time type checking between the object class and the
target class, in response to detecting that an indicator associated with the target
class in the object header has a second logic value.

10. The system of claim 7, wherein the processor further to
traverse a class hierarchy associated with the class of the object, in
response to determining that the first time type checking at the hotspot is failed.

5 11. The system of claim 7, wherein the processor further to
assert an indicator associated with the target class in the object header, in
response to determining in a second time type checking at the hotspot that the class
of the object and the target class match a type checking condition.

10 12. The system of claim 7, wherein the processor further to
return a signal indicating that the type checking is successful, in response
to determining that the class of the object and the target class match a
predetermined criterion.

15 13. The system of claim 7, wherein the memory further to save a beginning
address of a handle of the target class, and wherein the processor further to
detecting the hotspot by dynamic profiling.

14. A machine readable medium comprising a plurality of instructions that
20 in response to being executed result in a computing device
obtaining an object header from an object, and
checking a bit indicator in the object header to indicate a result of a first time
type checking at a hotspot between a class of the object and a target class specified
by the hotspot.

15. The machine readable medium of claim 14, wherein the machine readable medium further comprising instructions that in response to being executed result in the computing device skipping a second time type checking at the hotspot between the object class and the target class, in response to determining that the bit indicator is asserted to indicate a successful result.
16. The machine readable medium of claim 14, wherein the machine readable medium further comprising instructions that in response to being executed result in the computing device performing a second time type checking at the hotspot between the object class and the target class, in response to determining that the bit indicator is deasserted to indicate a failure result.
- 15 17. The machine readable medium of claim 14, wherein the machine readable medium further comprising instructions that in response to being executed result in the computing device detecting at the hotspot whether the object class and the target class match a type checking criterion, in response to determining that the bit indicator indicates 20 a failure result.

18. The machine readable medium of claim 16, wherein the machine readable medium further comprising instructions that in response to being executed result in the computing device asserting the bit indicator, in response to determining in the second time

5 type checking that the object class and the target class match a type checking criterion.

19. The machine readable medium of claim 16, wherein the machine readable medium further comprising instructions that in response to being executed

10 result in the computing device returning a signal indicating a result of the second time type checking.

20. The machine readable medium of claim 16, wherein the machine readable medium further comprising instructions that in response to being executed

15 result in the computing device throwing an exception, in response to determining that the second time type checking is successful.

21. The machine readable medium of claim 16, wherein the machine readable medium further comprising instructions that in response to being executed

20 result in the computing device pushing a result code on a stack to indicate whether the type checking is successful.

22. A system comprising,
a compiler to convert source code associated with a first time type checking
for an object into byte code;
a loader coupled with the compiler to load the byte code;
5 a dynamic compiler coupled with the loader to receive the byte code from
the loader, and to generate first native code associated with the first time type
checking based on the byte code; and
a profiler coupled with the dynamic compiler to detect a hotspot in the first
time type checking based on the first native code, and to return the hotspot to the
10 dynamic compiler.

23. The system of claim 22, wherein the dynamic compiler further to
regenerate second native code that calls a type checking function for the hotspot to
determine a type checking result between a class of the object and a target class
15 specified by the hotspot from an object header of the object.

24. The system of claim 22, wherein the dynamic compiler further to
regenerate second native code that calls a type checking function for the hotspot to
detect a type checking success between a class of the object and a target class
20 specified by the hotspot, in response to determining that an indicator associated
with the target class in an object header of the object has a first logic value.

25. The system of claim 22, wherein the dynamic compiler further to regenerate second native code that calls a type checking function for the hotspot to detect a type checking failure between a class of the object and a target class specified by the hotspot, in response to determining that an indicator associated with the target class in an object header of the object has a second logic value.

26. The system of claim 22, wherein the dynamic compiler further to regenerate second native code that calls a type checking function for the hotspot to skip type checking between a class of the object and a target class specified by the hotspot, in response to determining that an object header of the object indicates a type check success.

27. The system of claim 22, wherein the dynamic compiler further to regenerate second native code that calls a type checking function for the hotspot to perform type checking between a class of the object and a target class specified by the hotspot, in response to determining that an indicator associated with the target class in an object header of the object is deasserted.

28. The system of claim 22, wherein the dynamic compiler further to regenerate second native code that calls a type checking function for the hotspot to assert an indicator in an object header of the object, in response to a type checking success between a class of the object and a target class specified by the hotspot.

29. The system of claim 22, wherein the dynamic compiler further to regenerate second native code that calls a type checking function for the hotspot to traverse super classes of a class of the object to determine whether one of the super classes is the same as a target class as specified by the hotspot, in response to determining a type checking failure between the class of the object and the target class from an object header of the object.

30. The system of claim 22, wherein the dynamic compiler further to regenerate second native code that calls a type checking function for the hotspot to traverse a class hierarchy associated with a class of the object to determine whether a target class as specified by the hotspot is represented in the class hierarchy, in response to determining a type checking failure between the class of the object and the target class from an object header of the object.